Coding HTML with NVDA -Part 10(Form Elements)

# Introduction

Hey guys what’s up its your girl Thee Quinn here and I am back with another video in the html series. In today’s video, I will be showing you how to create form fields in html. But before we get into it, please be sure to like the video if you love the content, subscribe if you are new, and turn on my notification bell to be alerted whenever I post the next video. Also note that the instrumentals for this video have been provided by Inner Sanctum Entertainment Ltd. But without further ado, let’s get right into it.

# Start of tutorial

Before we begin, note that this video is only going to show you how to set out your forms on your webpage using html. As it relates to what the form will do after the user presses the submit button, that will not be covered in this video.

So a form is something that is typically used to collect input from the user. Forms are frequently found on web pages and have been used in many ways. I have seen them used for surveys, user login/registration, contact forms and even for calculators and converters. In html, the form element is what we use to create a form to collect user input. This user input will then be sent to the server for processing. The basic syntax for the form element is:

<form>

Form info

</form>

Now the form element is the main container for all the types of input elements for the form. These include edit boxes, radio buttons, checkboxes etc. To create fields to accept the input of the user, we would use the input tag. By default, using the input tag will create a text input field. In order to determine the type of input that will be accepted, we would use the type attribute in the starting input tag. The syntax is:

<input type="value">

Note that the input tag is an empty element, which means that it does not have a closing tag. Here are the most common values for the type attribute of the input tag:

* text: Displays a single-line text input field
* radio: Displays a radio button (for selecting one of many choices)
* checkbox: Displays a checkbox (for selecting zero or more of many choices)
* submit: Displays a submit button (for submitting the form)
* button: Displays a clickable button

You can get the complete list of all the possible type of form fields via the link in the section at the end of this document.

another attribute used within the input tag is the id attribute. The id attribute assigns an identifier to the input element. The id allows JavaScript to easily access the input element. It is also used to point to a specific id selector in a style sheet.

The input element should also contain the name attribute. This is required for the user’s input to be submitted. If the name attribute is omitted, the value of the input field will not be sent at all. Note that there are many other attributes for the input element. These include attributes to set the limit, disable the input etc. The complete list can be found via a link in the last section of this document.

# label

but before I get into demonstrating the use of the input tag, there is another important tag to learn. That is the label tag. This tag is what labels the form elements. So it will tell the user what they are required to input into the field. The label element is also useful for screen reader users, because the screen reader will read out loud the label when the user focuses on the input element. The label tag should also contain the for attribute. The value of this attribute should be equal to the value of the id attribute of the input tag, in order to bind them together. The complete syntax for the label tag is:

<label for="value">Text Goes Here</label>

Now let’s get into some demonstration.

# Single Line Text Input

now what if you want to create a single line field for text input? This can be used when asking for first name, username etc. This is when we will use the text as the value of the type attribute in the input tag. so the first thing we would need to do is to navigate to the index.html file that we have been working with, and open it with Notepad. I went ahead and deleted all of the code that we were working with in the last video. So right now, the only thing you should have in between the body tags is the heading. So we are going to go down into an empty line under the h1 tag. now if we are going to create a form, we are going to have to put it between the form tags. Now this form element will be within the body, so since we have been keeping consistent with 2 spaces, let’s put 2 spaces here. Now let’s put the opening form tag, and in the line under that, indenting by the same two spaces, let’s put the closing form tag at the same time. Now let’s get into a blank line in between the opening and closing form tags. Now since this text field that we are going to create is going to be within the form tag, and the form tag is already indented by 2 spaces, let’s indent it by 4 spaces. Now, before we put the input tag, we are going to have to put a label for it. So start typing your opening label tag, and before you close it, let’s put our for attribute. For this input field, I am going to ask the user for their first name. so I am going to give this a unique id because remember that this for attribute should match with the unique id that you will give to that input field that it will be labeling. So let’s call it “fname”. Now we can end the label tag and put our label, “First Name:”. Then we can put our ending label tag. now before we go into the next line, I must note. If you do not put a line break tag after the label tag, it will go directly beside the input field that you are using it to label. But in this case, I want it to go on top of this input field. So I am going to put my line break element which is the br tag. then we are going to enter to go into a new line, and indent it by 4 spaces as well. Now we are going to put our input tag. then type the input tag, setting the type as text, the id as “fname”, the name as “fname” and close the tag. This is the code that should be in between your body tags:

<form>

<label for="fname">First Name:</label><br>

<input type="text" id="fname" name="fname">

</form>

Now if you should save this, and open it in your browser, you should see the label and the edit box there. You can press enter on it to open it just like a regular edit box, and escape to close it. And if you should go to the top of the page, and press your letter E, it should take you right down to the edit box. Also note that the default width of an input field is 20 characters.

There is also another attribute that is used in the input tag. The value attribute is used to Specify the initial value of most of the elements. This is usually used on other types of input other than the text type, but I have seen persons use it by setting the initial text in the edit box to something like “enter text here” and then you would replace that text with your input. Now let’s move on to the next part of the demonstration.

# Radio button

The input element using radio as the type defines a radio button. Radio buttons allow a user to select ONE of a limited number of choices. For the radio button the value attribute contains a string that shows its value. It is not shown to the user but it is used to identify which radio button in a group is selected. All the radio buttons in a group will have the same value for the name attribute. this is so that you can easily determine which radio buttons belong to which group, and which one is selected. Also, in cases like these where each radio button will have their own label, the paragraph tag can be used to tell the user what to select. So, now let’s go back over to our index.html file.

So we are currently in the line with the input tag for the first name. let’s go to the end of the line, and put a line break tag after it. Then press enter to go into a new line and indent by 4 spaces. Now let’s use a paragraph tag to tell the user what this group of radio buttons is telling you to select. Lets put “please select a fruit”. Then after closing the paragraph tag, press enter to go into a new line and indent by 4 spaces again. now let’s start typing our input tag, putting a type of radio, apple for the id, fruits for the name, and apple for the value. Now, we want our label for the radio button to be right beside it, so therefore we will not put a line break at the end of this line. So let’s press enter to go into a new line, and indent by 4 spaces. Then let’s put the label tag with a value of apple for the for attribute, and label it “Apple”. Now, we want the radio button that is going to be after this to be in a new line, so let’s put the line break tag. then press enter to go into a new line and indent by 4 spaces. Next we are going to add another fruit. So type the input tag with the type as radio, id as melon, name as fruits (remember that we will give it the name as the rest of the radio buttons in this group), and value as melon. Then let’s enter to go into a new line, indent by 4 spaces again, and put the label tag for it. Then let’s follow these steps to create one more radio button for a fruit called pine. This should be your code in between the body tags:

<form>

<label for="fname">First Name:</label><br>

<input type="text" id="fname" name="fname"><br>

<p>Please select a fruit:</p>

<input type="radio" id="apple" name="fruits" value="apple">

<label for="apple">Apple</label><br>

<input type="radio" id="melon" name="fruits" value="melon">

<label for="melon">Melon</label><br>

<input type="radio" id="pine" name="fruits" value="pine">

<label for="pine">Pine</label><br>

</form>

Now if you should save it and open it in your browser, you should see the radio buttons. And would realize that if you should use either your spacebar or your enter key to check one of the options, then go and check another one, the one previously selected will be unselected. So this is how the radio buttons work, you are only allowed to select one option at a time. And if you should go to the top of the page and press the letter R, it will bring you straight to the first radio button. Now let’s move on to the next demonstration.

# Checkboxes

The input tag with a type of checkbox defines a checkbox. Checkboxes let a user select ZERO or MORE options from a group of items. So let’s go back to our index.html file and test this out. So while at the end of the line with the label for the last radio button we created, let’s press enter to go into a new line and indent by 4 spaces. Now let’s first put our paragraph to tell the user what we want them to enter. Let’s put “what devices do you use?”. Then press enter to go into a new line and keep consistent with the 4 spaces. Now let’s type our input tag with a type of checkbox, id as device1 since we are asking them to choose from a device, name as device1 (the name for the checkboxes do not have to be the same), and iPhone for the value. Then let’s go into a new line and indent by 4 spaces. Let’s put the label tag for it, setting the for attribute as device1. Let’s continue these steps to create two more checkboxes, with the second one being an Android, and the third being a Laptop. Your code between the body tags should be:

<form>

<label for="fname">First Name:</label><br>

<input type="text" id="fname" name="fname"><br>

<p>Please select a fruit:</p>

<input type="radio" id="apple" name="fruits" value="apple">

<label for="apple">Apple</label><br>

<input type="radio" id="melon" name="fruits" value="melon">

<label for="melon">Melon</label><br>

<input type="radio" id="pine" name="fruits" value="pine">

<label for="pine">Pine</label><br>

<p>What devices do you use?</p>

<input type="checkbox" id="device1" name="device1" value="iPhone">

<label for="device1">iPhone</label><br>

<input type="checkbox" id="device2" name="device2" value="android">

<label for="device2">Android</label><br>

<input type="checkbox" id="device3" name="device3" value="laptop">

<label for="device3">Laptop</label><br>

</form>

Now if you save this and open it in your browser, you should see the checkboxes there. You should also realize that if you use your spacebar to select one item, then another, both of them will remain selected. That is why I said checkboxes allow you to select more than one item at a time. And if you should go to the top of the page, and press your letter X, it will bring you to the first checkbox on the page. And that is it for checkboxes. Now let’s move on to another demonstration.

# Submit

The input element with a type of submit defines a button for submitting the form data to a form handler. The form-handler is typically a file on the server with a script for processing input data. The form-handler is specified in the form's action attribute. For example:

<form action="file\_path">

Now let’s go ahead and put in a submit button on our form. While at the end of the line with our last label (laptop), let’s press enter to go into a new line, and indent by 4 spaces. Then let’s type the input tag with a type of submit, and a value of submit. Your complete code in between the body tags should be:

<form>

<label for="fname">First Name:</label><br>

<input type="text" id="fname" name="fname"><br>

<p>Please select a fruit:</p>

<input type="radio" id="apple" name="fruits" value="apple">

<label for="apple">Apple</label><br>

<input type="radio" id="melon" name="fruits" value="melon">

<label for="melon">Melon</label><br>

<input type="radio" id="pine" name="fruits" value="pine">

<label for="pine">Pine</label><br>

<p>What devices do you use?</p>

<input type="checkbox" id="device1" name="device1" value="iPhone">

<label for="device1">iPhone</label><br>

<input type="checkbox" id="device2" name="device2" value="android">

<label for="device2">Android</label><br>

<input type="checkbox" id="device3" name="device3" value="laptop">

<label for="device3">Laptop</label><br>

<input type="submit" value="submit">

</form>

And if you should save it and open it in your browser, if you go to the end of the page, the button should be there. And you guys can go through all the other types of input on the link that I gave you earlier. But those are just the basic ones for this tutorial.

# Form attributes

There are also some other attributes that can be found in the starting form tag:

## Action

As seen before when creating the submit button, in the example, the action attribute was seen in the starting form tag. The action attribute defines the action to be performed when the form is submitted. Usually, the form data is sent to a file on the server when the user clicks on the submit button. If there is no action attribute specified in the form tag, the action is set to the current page

## Target

Another attribute used in the form tag is the target attribute. The target attribute specifies where to display the response that is received after submitting the form. The target attribute can have one of the following values:

* \_blank

This displays the response in a new window or tab

* \_self

The response is displayed in the current window

* \_parent

The response is displayed in the parent frame

* \_top

The response is displayed in the full body of the window

* framename

The response is displayed in a named iframe

If the target attribute is not specified it will resort to the default which is to open in the current window.

# Method

The method attribute specifies the HTTP method to be used when submitting the form data. The form data can be sent as URL variables (with method="get"),, or as HTTP post transaction (with method="post"). The default HTTP method when submitting form data is GET.

Notes on GET:

• Appends the form data to the URL, in name/value pairs

• NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)

• The length of a URL is limited (2048 characters)

• Useful for form submissions where a user wants to bookmark the result

• GET is good for non-secure data, like query strings in Google

Notes on POST:

• Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)

• POST has no size limitations, and can be used to send large amounts of data.

• Form submissions with POST cannot be bookmarked

Tip: Always use POST if the form data contains sensitive or personal information!

## Auto complete

The autocomplete attribute specifies whether a form should have autocomplete on or off. When autocomplete is on, the browser automatically completes values based on values that the user has entered before. To turn it on you would put “autocomplete=”on” in the starting form tag.

## The Novalidate Attribute

There is also the novalidate attribute. The novalidate attribute is a Boolean attribute. When present, it specifies that the form data should not be validated when submitted. This attribute has no value, so you would just put the word novalidate in the starting form tag.

More attributes can be found via the link in the last section of this document. There are also other elements other than the input and the label that can be used in the form element. These include elements to create drop down lists and multi line text inputs. You can find these via the link in the last section of this document.

# Links

1. Input types link:

<https://www.w3schools.com/html/html_form_input_types.asp>

1. Attributes for the input element:

https://www.w3schools.com/html/html\_form\_attributes.asp

1. Link to more form tag attributes:

<https://www.w3schools.com/tags/tag_form.asp>

1. link to other form elements:

<https://www.w3schools.com/html/html_form_elements.asp>